www.sqlbi.com

a brand of Loader

Microsoft Partner Gold Data Platform Gold Business Intelligence

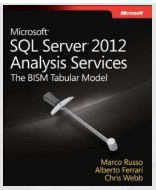
MVP Microsoft Most Valuable Professional

SSAS MAESTRO by Microsoft

alberto.ferrari@sqlbi.com

Time Intelligence in DAX

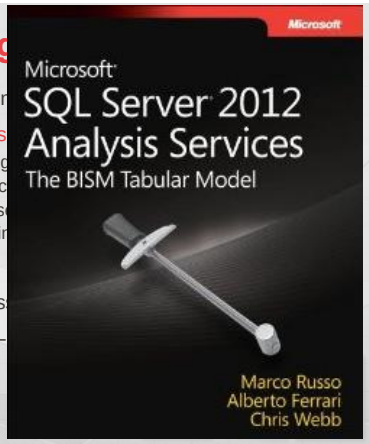
Who's that guy?



- BI Expert and Consultant
- Founder of www.sqlbi.com
  - Problem Solving
  - Complex Project Assistance
  - Data Warehouse Assessments and Development
  - Courses, Trainings and Workshops
- Book Writer
- Microsoft Business Intelligence Partner
- SSAS Maestros – MVP – MCP

Who's that g

- BI Expert and Cor
- Founder of www.s
  - Problem Solving
  - Complex Projec
  - Data Warehous
  - Courses, Trainir
- Book Writer
- Microsoft Busines
- SSAS Maestros –



What is Time Intelligence?

- Many different topics in one name
  - Year To Date
  - Quarter To Date
  - Running Total
  - Same period previous year
  - Working days computation
  - Fiscal Year
- Well…seems to be anything related with time, handled in an intelligent way ☺

What do you need to know?

- Not really much
  - DAX basics
  - CALCULATE basics
- If not, add 100 to the session level
- Topic is DAX, we are going to use PowerPivot

Calendar Table

- First of all we need some data
- And a calendar table
  - Built in Excel
  - Or in a SQL Table
- Calendar table properties
  - All dates should be present
  - No holes
  - Otherwise time intelligence will not work

Year To Date: the easy way

TOTALYTD: life is easy

```
SalesAmountYTD :=

TOTALYTD(
    SUM (Sales[SalesAmount]),
    Calendar[FullDate]
)
```

Mark as Date Table

- In Tabular, use Mark as Date Table
- Set the column containing the date
- Needed to make time intelligence works
- Used by Power View as metadata information
- Many tables can be marked as date table

Set Sorting Options

- Month names do not sort alphabetically
  - April is not the first month of the year
- Use Sort By Column
- Set all sorting options in the proper way
- Beware of sorting granularity
  - 1:1 between names and sort keys

Use The Right Parameter

The parameter is the Date column in the Calendar Table
Not the SalesOrderHeader[OrderDate]
Otherwise, you get wrong results

```
LineTotalYTD :=

TOTALYTD(
    SUM (Sales[SalesAmount]),
    Sales[OrderDate]
)
```

Handling Fiscal Year

The last, optional, parameter is the end of the fiscal year
Default: 12-31 (or 31/12 - locale dependent)

```
SalesAmountYTD :=

TOTALYTD (
    SUM (Sales[SalesAmount]),
    Calendar[FullDate],
    "06-30"
)
```

Year To Date: the DAX way

- DATESYTD: Returns a set of dates, from the start of the year up to the parameter date
- CALCULATE: Creates a filter context and performs the SUM operation

```
SalesAmountYTD :=

CALCULATE (
    SUM (Sales[SalesAmount]),
    DATESYTD (Calendar[Date])
)
```

Running Total

- Running total, as most of the more complex time intelligence aggregations, needs the CALCULATE version, because there is no syntax sugaring here

```
SalesAmountRT :=

CALCULATE (
    SUM (Sales[Amount]),
    FILTER (
        ALL (Calendar),
        Calendar[FullDate] <= MAX (Calendar[FullDate])
    )
)
```

Same Period Last Year

Same period in previous year, CALCULATE is needed
Specialized version of DATEADD

No hierarchies, here, different technique when compared with Multidimensional

```
SalesSPLY :=

CALCULATE (
    SUM (Sales[SalesAmount]),
    SAMEPERIODLASTYEAR (Calendar[FullDate])
)
```

DATEADD

Similar to SAMEPERIODLASTYEAR, used to calculate different offsets

```
SalesAmountlSPLY :=

CALCULATE(
    SUM (Sales[SalesAmount]),
    DATEADD (Calendar[FullDate], -1, YEAR)
)
```

PARALLELPERIOD

Returns a FULL period of dates shifted in time

The whole period is returned, regardless dates in the first parameter, very useful to compute percentages

```
SalesPPLY :=

CALCULATE(
    SUM (Sales [LineTotal]),
    PARALLELPERIOD (Calendar[FullDate], -1, YEAR)
)
```

Period Table Approach

- Many Time Intelligence Aggregations
  - Many measures
  - User Experience might not be the best
- Tool Table (tool dimension in UDM)
  - Reduces the number of measures
  - Base measures can be hidden
- Drawbacks
  - The code is harder to debug

Multiple Calendar Tables

- Calendar is often a role dimension
  - Many roles for a date
  - Many calendar tables
- How many calendar tables?
  - Use many, only if needed by the model
  - Try to use only one table
  - Many calendars leads to confusion
  - And issues when slicing
- Use proper naming convention

Counting Working Days

- How many working days in a date range?
  - Easily solved with Calendar table
  - Define a new column «WorkingDays»
  - Aggregate with SUM
- Handles any date range
- Works on any periods
- No separation between fact tables and dimensions in Tabular

Check Delayed Orders

- How many orders were delayed?
  - Easy: «ShipDate Greater Than DueDate»
- How many working days of delay?
  - New Calculated Column
  - WorkingDayNumber
    - Incremental value from the start of table
    - Delta can be computed as difference

sqlbi

Handling DateTime

- If time is a useful information
- Separate Date from Time
  - Date part → Calendar Table
  - Time part → Time Table
- Reduces distinct values
- Makes analysis much easier

sqlbi

Semi Additive Measures

- Additive Measure
  - SUM over all dimensions
- Non Additive Measure
  - Different function over all dimensions
  - Example: average of the sale price
- Semi Additive Measure
  - SUM over some dimensions
  - Different function over other dimensions
  - Time is the standard exception for aggregations
  - Examples
    - Warehouse stocking
    - Current account balance

sqlbi

Current Account Balance

Name	Occupation	Country	Date	Quarter	Balance
Katie Jordan	Farmer	USA	1/1/2010	Q1/2010	1,687.50
Luis Bonifaz	IT Consultant	Argentina	1/1/2010	Q1/2010	1,470.00
Maurizio Macagno	IT Consultant	Italy	1/1/2010	Q1/2010	1,500.00
Katie Jordan	Farmer	USA	2/1/2010	Q1/2010	2,812.50
Luis Bonifaz	IT Consultant	Argentina	2/1/2010	Q1/2010	2,450.00
Maurizio Macagno	IT Consultant	Italy	2/1/2010	Q1/2010	2,500.00
Katie Jordan	Farmer	USA	3/1/2010	Q1/2010	3,937.50
Luis Bonifaz	IT Consultant	Argentina	3/1/2010	Q1/2010	3,430.00

Sum of Balance	Column Labels			
Row Labels	Katie Jordan	Luis Bonifaz	Maurizio Macagno	Grand Total
Q1/2010	8,437.50	7,350.00	7,500.00	23,287.50
1/1/2010	1,687.50	1,470.00	1,500.00	4,657.50
2/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
3/1/2010	3,937.50	3,430.00	3,500.00	10,867.50
Q2/2010	6,975.00	6,076.00	6,200.00	19,251.00
4/1/2010	2,250.00	1,960.00	2,000.00	6,210.00
5/1/2010	2,025.00	1,764.00	1,800.00	5,589.00
6/1/2010	2,700.00	2,352.00	2,400.00	7,452.00
Q3/2010	11,475.00	9,996.00	10,200.00	31,671.00
7/1/2010	3,600.00	3,136.00	3,200.00	9,936.00
8/1/2010	5,062.50	4,410.00	4,500.00	13,972.50
9/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
Q4/2010	6,862.50	5,978.00	6,100.00	18,940.50
10/1/2010	2,250.00	1,960.00	2,000.00	6,210.00
11/1/2010	2,081.25	1,813.00	1,850.00	5,744.25
12/1/2010	2,531.25	2,205.00	2,250.00	6,986.25
Grand Total	33,750.00	29,400.00	30,000.00	93,150.00

- Month level correct
- Quarter level wrong
- Year level wrong

sqlbi

SemiAdditive Measures

CALCULATE: to set the filter
LASTDATE: to find the last child

```
LastBalance :=

CALCULATE (
    SUM (Balances[Balance]),
    LASTDATE (Date[Date])
)
```

Moving Annual Total

Moving window from the current date back one year

```
CALCULATE(
    SUM( SalesOrderDetail[LineTotal] ),
    DATESBETWEEN(
        OrderDate[Date],
        NEXTDAY(
            SAMEPERIODLASTYEAR(
                LASTDATE( OrderDate[Date] )
            )
        ),
        LASTDATE( OrderDate[Date] )
    )
)
```

Querying with Time Intelligence

Using SUMMARIZE on the fact table yiedls an error, because dates in the fact table usually contain holes

```
DEFINE
    MEASURE 'Internet Sales'[PY Sales] =
        CALCULATE(
            [Internet Total Sales],
            SAMEPERIODLASTYEAR( 'Date'[Date] )
        )

EVALUATE

SUMMARIZE(
    'Internet Sales',
    'Date'[Calendar Year],
    'Date'[Month],
    'Date'[Month Name],
    "Sales", [Internet Total Sales],
    "PY Sales", [PY Sales]
)
```

Querying with Time Intelligence

The solution is to use the calendar table as the primary one, changing the shape of the query.

```
DEFINE
    MEASURE 'Internet Sales'[PY Sales] =
        CALCULATE(
            [Internet Total Sales],
            SAMEPERIODLASTYEAR( 'Date'[Date] )
        )
EVALUATE
ADDCOLUMNS(
    FILTER(
        SUMMARIZE(
            'Date',
            'Date'[Calendar Year],
            'Date'[Month],
            'Date'[Month Name],
            "Sales", [Internet Total Sales]
        ),
        [Sales] <> 0
    ),
    "PY Sales", [PY Sales]
)
```

Conclusions

- Time Intelligence is pretty easy
  - Many built-in functions
  - Calendar table is very important
- Based on DAX formulas
  - Not no hierarchies
- Formulas are sometime complex
- But you can author very powerful calculations
- Thus… learn DAX ☺

sqlbi

Thank you!

Check daily our new articles on
www.sqlbi.com

Oh, just another thing…

Are you sure to get the best out of your analytical environment?

Ask to SQLBI, we offer many services to help you:


Consulting


Assessment


Outsourcing


Technical Fellowship

Find out more on
www.sqlbi.com/consulting